Let's assume the array is supposed to have numbers from 1 to n, but one number is missing, and one number is repeated. Let the missing number be M and the repeated number be R.

Step-by-Step Derivation:

1. **Sum of the First n Natural Numbers:**

   The sum of the first n natural numbers is given by:

   $S_n = \frac{n(n+1)}{2}$

2. **Sum of the Squares of the First n Natural Numbers:**

   The sum of the squares of the first n natural numbers is given by:

   $S_{n^2} = \frac{n(n+1)(2n+1)}{6}$

3. **Calculate the Actual Sum and Sum of Squares:**

   Let the actual sum of the elements in the array be S_{actual} and the actual sum of the squares of the elements be $S_{\text{actual squares}}$.

4. **Form the Equations:**

   Since the sum S_{actual} differs from S_n by $R - M$:

   $S_{\text{actual}} = S_n + R - M$

   Similarly, the sum of the squares $S_{\text{actual squares}}$ differs from S_{n^2} by $R^2 - M^2$:

   $S_{\text{actual squares}} = S_{n^2} + R^2 - M^2$

5. **Substitute the Known Values:**

   Using the values S_n and S_{n^2}, substitute them into the equations:

   $S_{\text{actual}} = \frac{n(n+1)}{2} + R - M$

   $S_{\text{actual squares}} = \frac{n(n+1)(2n+1)}{6} + R^2 - M^2$

6. **Simplify the Equations:**

   From the first equation, we get:

   $R - M = S_{\text{actual}} - \frac{n(n+1)}{2} \quad \text{(Equation 1)}$

   From the second equation, since $R^2 - M^2 = (R - M)(R + M)$:

$$R^2 - M^2 = S_{\text{actual squares}} - \frac{n(n+1)(2n+1)}{6}$$

Substituting $R - M$ from Equation 1:

$$(R - M)(R + M) = S_{\text{actual squares}} - \frac{n(n+1)(2n+1)}{6}$$

Let $D = R - M$.

Then:

$$D(R + M) = S_{\text{actual squares}} - \frac{n(n+1)(2n+1)}{6}$$

Hence:

$$R + M = \frac{S_{\text{actual squares}} - \frac{n(n+1)(2n+1)}{6}}{D}$$

7. **Solve for R and M:**

   Now, you have two equations:

   $$R - M = D \quad (\text{Equation 1})$$

   $$R + M = \frac{S_{\text{actual squares}} - \frac{n(n+1)(2n+1)}{6}}{D} \quad (\text{Equation 2})$$

   Adding and subtracting these equations will give you R and M:

   Adding Equation 1 and Equation 2:

   $$2R = D + \frac{S_{\text{actual squares}} - \frac{n(n+1)(2n+1)}{6}}{D}$$

   $$R = \frac{D + \frac{S_{\text{actual squares}} - \frac{n(n+1)(2n+1)}{6}}{D}}{2}$$

   Subtracting Equation 1 from Equation 2:

   $$2M = \frac{S_{\text{actual squares}} - \frac{n(n+1)(2n+1)}{6}}{D} - D$$

   $$M = \frac{\frac{S_{\text{actual squares}} - \frac{n(n+1)(2n+1)}{6}}{D} - D}{2}$$

This derivation allows you to find the missing number M and the repeated number R in the array.

The Tortoise and Hare cycle detection algorithm, also known as Floyd's Cycle Detection Algorithm, is a pointer algorithm that uses two pointers to detect a cycle in a sequence.

It’s commonly used to detect cycles in linked lists but can be applied to any scenario where cycle detection is required.

Derivation and Explanation

Problem Statement

Given a sequence of elements, we want to determine whether there is a cycle, and if so, find the entry point of the cycle.

Concept

The algorithm uses two pointers:

1. **Tortoise (slow pointer)**: Moves one step at a time.
2. **Hare (fast pointer)**: Moves two steps at a time.

Steps of the Algorithm

1. **Initialization**:
   - Start both pointers at the beginning of the sequence.
2. **Phase 1: Cycle Detection**:
   - Move the Tortoise by one step and the Hare by two steps in each iteration.
   - If there is a cycle, at some point, the Tortoise and Hare will meet inside the cycle.
   - If the Hare reaches the end of the sequence (null), then there is no cycle.
3. **Phase 2: Finding the Entry Point**:
   - Once a cycle is detected (Tortoise and Hare meet), move the Tortoise to the start of the sequence.
   - Keep the Hare at the meeting point.
   - Move both pointers one step at a time; the point where they meet again is the entry point of the cycle.

Detailed Derivation

Phase 1: Cycle Detection

1. **Initialization**:
   - Let $\text{Tortoise} = \text{head}$
   - Let $\text{Hare} = \text{head}$
2. **Move Pointers**:
   - Move the Tortoise one step at a time: $\text{Tortoise} = \text{Tortoise.next}$
   - Move the Hare two steps at a time: $\text{Hare} = \text{Hare.next.next}$
3. **Check for Cycle**:

- If Hare meets $\mathrm{Tortoise}$, a cycle is detected.
- If Hare reaches the end of the sequence, there is no cycle.

Phase 2: Finding the Entry Point

1. **Initialization**:
   - Move the Tortoise to the start of the sequence: $\mathrm{Tortoise} = \mathrm{head}$
   - Keep the Hare at the meeting point.
2. **Move Pointers**:
   - Move both pointers one step at a time:
     - $\mathrm{Tortoise} = \mathrm{Tortoise.next}$
     - $\mathrm{Hare} = \mathrm{Hare.next}$
3. **Finding the Entry Point**:
   - The point where Tortoise and Hare meet again is the entry point of the cycle.

Mathematical Explanation

Let's assume there is a cycle in the sequence, and let:

- L be the length from the start of the sequence to the start of the cycle.
- C be the length of the cycle.
- k be the distance from the start of the cycle to the meeting point of Tortoise and Hare.

When the Tortoise and Hare meet, the distance covered by the Tortoise is: $L + k$

The distance covered by the Hare is twice that of the Tortoise: $2(L + k)$

However, the Hare has also completed an extra cycle: $L + k + mC$

Where m is the number of cycles completed by the Hare.

Equating the distances: $2(L + k) = L + k + mC$

Solving for L: $L + k = mC$

This means the distance from the start to the start of the cycle L is a multiple of the cycle length C.

When we reset the Tortoise to the start of the sequence and move both pointers one step at a time, they will meet at the start of the cycle after L steps, which confirms the entry point of the cycle.

Example

Consider the sequence $[3, 2, 0, -4]$ where the position 1 is the start of the cycle:

1. **Phase 1: Cycle Detection**
    - Initialize: Tortoise = 3, Hare = 3
    - Step 1: Tortoise = 2, Hare = 0
    - Step 2: Tortoise = 0, Hare = -4
    - Step 3: Tortoise = -4, Hare = 0 (Cycle detected)
2. **Phase 2: Finding the Entry Point**
    - Initialize: Tortoise = 3, Hare = 0
    - Step 1: Tortoise = 2, Hare = 2 (Entry point found)

This example demonstrates the Tortoise and Hare algorithm successfully detecting the cycle and finding its entry point.